Current Issues of Malicious Domains Blocking

Stanislav Špaček*†, Martin Laštovička*† Martin Horák* and Tomáš Plesník*
*Masaryk University, Institute of Computer Science, Brno, Czech Republic
†Masaryk University, Faculty of Informatics, Brno, Czech Republic
Email: {spaceks|lastovicka|horak|plesnik}@ics.muni.cz

***Abstract*—Cyberattackers often use the Domain Name System (DNS) in their activities. Botnet C&C servers and phishing websites both use DNS to facilitate connection to or from its victims, while the protocol in its basic form does not contain any security countermeasures to thwart such behavior. In this paper, we examine capabilities of a DNS firewall that would be able to filter access from the protected network to known malicious domains on the outside network. Considering the needs of Computer Security Incident Response Teams (CSIRTs), we formulated functional requirements that a DNS firewall should fulfill to fit the role of a cybersecurity tool. Starting from these requirements, we developed a DNS firewall based on the DNS Response Policy Zones technology, the only suitable open source technology available yet. However, we encountered several essential limitations in the DNS RPZ technology during the testing period. Still, our testing results show that simple DNS firewall can prevent attacks not detected by other cybersecurity tools. We discuss the limitations and propose possible solutions so that the DNS firewall might be used as a more complex cybersecurity tool in the future. Lessons learned from the deployment show that while the DNS firewall can indeed be used to block access to malicious domains, it cannot yet satisfy all the requirements of cybersecurity teams.**

I. Introduction

The Domain Name System (DNS) is being widely taken advantage of by various cyberattackers for years now. With no level of security control, the DNS translates a known malicious domain when queried as well as a benign one. The attackers are aware of it and use domain names as aliases for their Command and Control (C&C) servers. In these cases, simple blocking of attacker's IP address is not enough, as the attackers have techniques that allow them to rapidly change IP addresses while their domain name remains the same. These techniques render IP blocking ineffective [1]. Similar cases when DNS unsuspectingly translates a malicious domain name happen during phishing campaigns that employ typo-squatting technique when attackers register domain names similar to well known and trusted domains in an attempt to create a false sense of security in their victims [2]. A promising adaptation to this behavior would be an *intelligent* local DNS resolver, a DNS firewall, able to distinguish whether a malicious or a benign domain is being translated and to prevent potentially harmful connections.

Aside from the cybersecurity issues the DNS firewall would help to solve, its deployment might also be driven by the need to fulfill legal requirements. In this role, the DNS firewall poses as a de facto form of Internet censorship [3]. On the one hand, there are countries with authoritarian governments such as China, North Korea, Iran, etc. which implement DNS blocking to block websites belonging to their political opposition, independent media, etc. On the other hand, there are non-authoritarian countries which also have motives for using DNS filtering. Rationales of these countries can be blocking of obscene content, defamation, harassment, state security concerns, or intellectual property protection. For example, Internet Service Providers (ISPs) in the Czech Republic are obliged to block websites providing non-permitted Internet gambling according to the Act On Gambling [4].

In this paper, we focus on technological aspects of this measure, having in mind that a DNS firewall must also provide supplementary functions like management access and auditing to fit among other cybersecurity tools. In cooperation with our cybersecurity incident handlers, we formulated functional requirements that the DNS firewall should fulfill. These requirements should cover the needs of any Computer Security Incident Response Team (CSIRT):

- **Domain Blocking** – The DNS firewall must be able to block translation of blacklisted malicious domains. The blocking of domain translation effectively disrupts C&C infrastructure of botnets and prevents users from interacting with malicious websites. It should also provide a management interface for easy access and configuration.
- **User Notifying** – When users try to interact with a blocked domain, it is not sufficient to just block the DNS translation. The users should be informed about why the domain is blocked but contacting them outside of the DNS firewall system is often not a trivial task. The DNS firewall could immediately redirect them to an informational landing page, so the DNS filtering remains transparent and educates users to be more careful in the future.
- **Event Logging** – The DNS firewall must be able to log attempts to access blacklisted domains. Analysis of this log allows detecting anomalies that might indicate an attack or misconfiguration in the local network.
- **Domain Blacklist Sharing** – Similarly to fast IP switching, the attackers can quickly change domain names of their servers. If every DNS firewall acted separately, the reaction window would be too large for efficient attack prevention. To minimize the size of this window, organizations using DNS firewall should either share their blacklist or subscribe to updates from a central authority.

We designed a DNS firewall system based on the above requirements, using the only open source technology currently available – DNS Response Policy Zones (DNS RPZ) [5]. We tested the firewall in both isolated environment and real network traffic on our campus network. We discovered that the DNS RPZ technology suffers from serious limitations during our testing, most of them arising from the fact that DNS RPZ uses techniques similar to DNS session hijacking. We have recently deployed a DNS firewall that fulfills the requirements of domain blocking, event logging and partly user informing, but fails in blacklist sharing. The data we gathered so far shows that the DNS firewall complements other cybersecurity tools and detects anomalies that would otherwise be overlooked. We present the limitations we identified and propose our solutions to promote discussion on the topic. We believe that if these issues are resolved, the DNS firewall will prove a valuable tool in the field of cyberattack prevention.

The rest of this paper is organized as follows. Section II mentions other works and projects concerning DNS filtering. Section III describes the specifications of the DNS RPZ technology. Section IV describes our implementation of DNS firewall for the Cybersecurity Incident Response Team of Masaryk University (CSIRT-MU). Section V highlights the open issues of current DNS firewall technology we encountered and Section VI concludes with lessons learned.

II. Related Work

The idea to use the DNS filtering to combat online threats comes from a blog post in 2010 by one of the authors of the DNS, Paul Vixie [6]. Vixie claimed that most of the newly registered domain names at the time were malicious and proposed an open source technology for domain blacklisting called DNS Response Policy Zones.

The DNS RPZ is currently the only existing open source technology for DNS firewalls usable on the institutional level. Only a handful of papers was published in this area over the last seven years. A case study was conducted in 2012, examining the impact of DNS RPZ on network traffic security [7]. In 2013, a technical report was published by Connery [8], containing information on how to correctly implement DNS RPZ as a security measure and join the DNS RPZ community. Another report was published by Connery [9] in the same year, providing an overview of the DNS RPZ and demonstrating several use-cases. This report also described possible future directions of DNS RPZ technology development.

Approximately at the same time as the DNS firewall was being considered as a cybersecurity tool, the discussion began about the usage of DNS filtering to block domains that host content infringing on intellectual property rights. Crocker et al. argue that state-wide enforced DNS filtering would be easy to circumvent, might cause collateral damages, and that it is not compatible with the authenticated and secure version of DNS – DNSSEC [10]. French top-level domain name operator Afnic confirms the relevance of these arguments in their own report [11]. We believe that most of these issues can be solved when the DNS firewall is used on a smaller scale, e. g. in an organization. The circumvention problem might be tackled by transparent blocking of exclusively malicious targets while informing users about the potential risks and giving them no incentive to opt out.

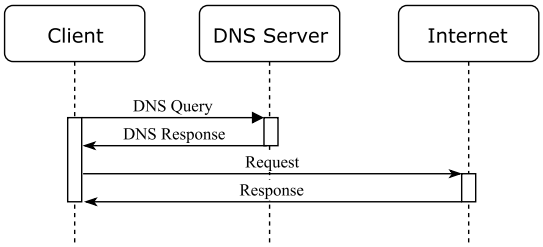

Fig. 1. Standard query processing in DNS

To reliably block only malicious domains, they should be either checked manually, or an accurate automatic procedure could be employed. Research is being done in the field of timely automatic recognition of malicious domain names. Prakash et al. propose an algorithm for predictive generation of malicious URLs used in phishing campaigns [12]. Proactive blacklisting of malicious domain names is also the topic of a study conducted by Felegyhazi et al. [13]

III. DNS RPZ Technology

The DNS RPZ technology may be considered a de facto standard for exerting control over DNS communication for the cybersecurity cause. It was designed by Paul Vixie, one of the people behind the DNS protocol itself. The technology is defined in an RFC draft where last changes were committed in June 2018 [5]. It is implemented in the widely used Berkeley Internet Name Domain (BIND) DNS server since version 9.8, so it can be set up on any BIND-based open source DNS resolver. Beside open source applications, the DNS RPZ is also used in several commercial DNS firewalls as well. For example in BlueCat DNS [14], and DNS firewall by InfoBlox [15].

The DNS RPZ blacklist corresponds to a DNS zone file commonly used in BIND. The rules contained therein are written in the form of zone file entries. However, the DNS RPZ defines its specific keywords in two categories – actions and triggers. Each rule consists of exactly one action and one trigger. For example, a DNS RPZ rule might have the following syntax:

malicious.domain.com IN CNAME rpz-drop.

Where *malicious.domain.com* is the trigger, *rpz-drop.* is the action, and *IN* and *CNAME* are regular DNS zone file parameters. When processing a query, the DNS RPZ sequentially goes through all the rules in its blacklist. If a matching trigger is found, a firewall action triggers, otherwise the DNS communication is not tampered with. The difference of how a query is handled in DNS and DNS RPZ is illustrated on Figures 1 and 2, where the DNS RPZ provides a modified reply to redirect user on a safe page.

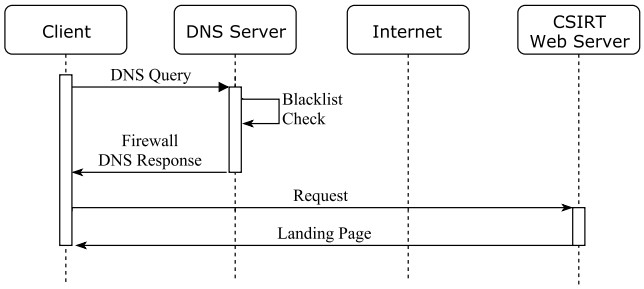

Fig. 2. Query processing in the DNS RPZ

The DNS RPZ technology defines five triggers that may invoke a rule based on different conditions:

- **QNAME Trigger** – It is based on the name of the queried domain. Allows the usage of a wildcard character to be applied to all subdomains.
- **IP Trigger** – Checks the content of a DNS reply for the returned IP address. If the returned IP address is present on the blacklist, the rule applies.
- **CLIENT IP Trigger** – The rule applies if the IP address of the querying DNS client is present on the blacklist.
- **NSDNAME Trigger** – This trigger allows issuing a modified DNS reply if the domain name of the delegated DNS server is present on the blacklist.
- **NSIP Trigger** – This trigger allows issuing a modified DNS reply if the IP address of the delegated DNS server is present on the blacklist.

The DNS RPZ defines six following actions that alter the DNS reply in different ways.

- **NXDOMAIN** – Blocks the communication. The client is returned a reply that the queried domain name does not exist.
- **NODATA** – Blocks the communication. The client is returned a reply that the queried domain name exists, but its IP address is not contained in the reply.
- **PASSTHRU** – Allows the communication as in standard DNS, but creates a log entry in the DNS firewall log.
- **DROP** – Blocks the communication. Both the query and the reply are discarded. The user gets no answer besides connection timeout.
- **TCP-Only** – Requires the client to resend the usual UDP DNS query over TCP. It is designed to mitigate the impact of DDoS attacks on DNS RPZ servers.
- **Local Data** – Redirects the communication. The communication is redirected to a specified domain name or IP address.

The above-specified triggers and actions may be mixed into specific rules. The action segment of the rule allows to either block the client connection or redirect it on a different server. The PASSTHRU action is reserved for special exceptions that should be let through regardless of whether they appear later on the blacklist. This way, a rule can be created that enables access to *secure.example.com* while *example.com* and all of its other subdomains remain blocked. It is also useful for whitelisting critical domains so that they can't be blocked by mistake in the future.

The triggers provide much variability in rule construction. Aside from blocking access to malicious domains, they allow blocking all DNS queries from a specific client device or discard DNS replies from a malicious outside DNS server. By using CLIENT IP Trigger, we can redirect all DNS queries from an infected device on an internal network to a landing page with instructions on how to proceed to purge the infection. This way, infected devices can be easily quarantined. By cutting DNS connection, they get isolated from attacker's C&C servers and are unable to cause further mischief [6]. NSDNAME and NSIP provide means to block replies from specific external DNS servers. However, caution needs to be exercised when blocking a DNS server. Authoritative nameservers usually translate more than one domain. If an authoritative DNS server is blocked, all the domains it manages will be inaccessible. If the block is a false positive, significant damage to traffic on the protected network may be caused.

If the DNS firewall blacklist is constructed manually, the probability of false positives is not high, as each entry might be manually examined. However, such procedure is highly inefficient and requires constant work. The DNS RPZ contains a method for blacklist sharing, so any malicious domain identified by one organization can be shared with others. The sharing method uses existing protocol for Incremental Zone Transfer (IXFR), as DNS RPZ blacklist behaves as a DNS zone file. The sharing system is designed so that the regular blacklist consumers would subscribe to several large producers while sustaining and sharing their own private blacklist [16].

IV. DNS Firewall Implementation

Starting from the requirements on DNS firewall we identified, we have designed and deployed the DNS firewall on a large-scale campus network.

A. Module Implementation

The DNS RPZ provides only the means to block access to malicious domains. Most importantly, it lacks any administrative interface. This means that any configuration must be initially done directly, by accessing blacklist files and editing the rules. We implemented several modules to support DNS RPZ firewall – administrative GUI with remote access to firewall blacklist, logging module and a landing page to serve as a target for user redirection.

Our DNS firewall system has a single main purpose – blocking access to known malicious domains. We decided that we will only use user redirection as opposed to simple domain blocking. When operating a DNS firewall for a large network with tens of thousands of users, the efficiency of simple blocking is limited if the blocked domain is accessed directly by users through the web browser. With no way of knowing the cause of the domain suddenly being unavailable, users would search for other ways to access it or generate tickets for their helpdesk [11]. We believe that redirecting

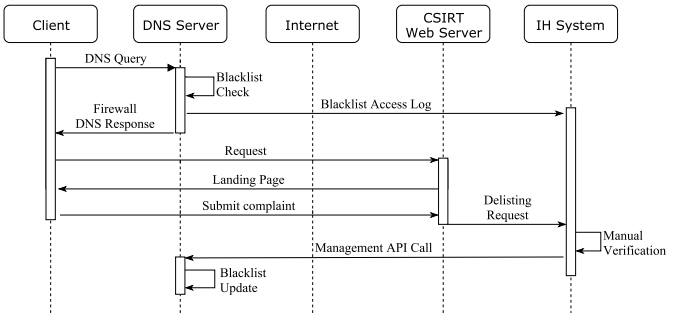

Fig. 3. Query processing in the DNS RPZ with a management interface and a logging module.

users is a much better approach. We designed a secure landing page that presents the user with information about the security incident that just occurred and a course of action to appeal for lifting the block. It should be mentioned that the landing page server can only answer users visiting it over HTTP or HTTPS. If the user initiated a connection over other protocol, it is refused and the redirection works similarly to simple domain blocking.

To maintain the awareness of the DNS firewall operation, we implemented an event logging system. Every attempt at accessing a blocked domain is logged through the firewall connector into incident handling system. Similarly, all administrative actions taken in the firewall are logged. This way, our handlers have all relevant information at hand if any users decide to contact them concerning specific security incident. Retroactive or even-real time log analysis is also possible. The analysis might point to anomalies that will need to be interpreted. These anomalies encompass infected devices trying to contact their C&C servers, users that fell victim to phishing campaigns and similar incidents. Currently, there are attempts at DNS RPZ log interpretation [17], but further research in anomaly detection will be required.

To more concisely demonstrate our DNS firewall operation, we present an example of DNS query processing in Figure 3. In this example, the client sends a standard query to the DNS server. However, the domain name in the query is malicious and is listed on the DNS firewall blacklist. The DNS server logs this communication attempt and responds with a modified DNS reply. The client is then redirected to a landing page, where the original domain name is displayed along with the reason of why it was blocked together with a contact form to the security team. In this case, the user chooses to complain to the handlers and requests to lift the block. The request is automatically sent from the web server to the incident handling system, where the handlers manually verify this claim. If they evaluate the domain to be benign, they unblock it through API directly on the DNS server.

B. Firewall Deployment

We have created a prototype of the system and deployed it on our campus network for several rounds of testing. All of the key system parts were implemented and interconnected – a DNS RPZ firewall with an API for remote access, administrative GUI with log processing module and a webserver with a firewall landing page.

First, we ran a test to estimate the importance of the DNS firewall on a large scale network. For this purpose, the firewall was temporarily set to only log accesses to blacklisted domains without blocking them. Our campus network is a quickly changing environment with around 43 000 static and mobile devices. We manually created a blacklist of 38 known malicious domains that figured in security incidents in our network during January 2018. The incidents were detected by a proprietary anomaly detection system deployed in our infrastructure and we analyzed the traffic to discover the target domains. Since all the domains were manually examined, it is safe to assume the blacklist did not contain any false positives.

We fitted our DNS firewall with the blacklist and logged events on our network for one month since 12th of February to 12th of March 2018. During this period, the firewall detected 217 accesses to the blacklisted domains by 23 distinct IP addresses. We believe that this number is high enough to affirm the importance of the use of DNS firewall along other cyberattack countermeasures.

After the initial importance test, the prototype entered another testing stage. The DNS firewall was tested for implementation faults in an isolated environment. When we finished testing of the modules and their interconnection, we switched to testing of various scenarios for chosen groups of users and firewall administrators. Finally, we deployed the prototype on our main DNS resolvers for the whole campus network. Unfortunately, we encountered a major limitation of the DNS RPZ in the final firewall testing stage. If users browse websites over HTTPS, it is impossible to redirect them to the landing page as the browser identifies it as an attack attempt. This is still an open issue and it is more closely described in Section V.

Until this issue is resolved, the DNS RPZ-based firewall is restricted only to simple domain blocking, and users must be informed of the fact using other channels. We are currently using our web pages to display the list of blocked domains for regular users and local network administrators. We are also considering the possibility of sending e-mails to users that attempted to access a blocked domain. The e-mail can contain all the information formerly accessible through the landing page. However, both these processes are not ideal and come with a caveat. We believe that the direct link between the attempted access and the reason for blocking, which was provided by user redirection, is now too loose.

Another issue arose with the implementation of the blacklist sharing function. Several commercial subjects exist, that provide their DNS RPZ blacklists in a paid subscription model, often along with other security services. However, this model is not feasible for CSIRTs that develop and rely on their own security tools. Besides subscribing to a paid service, it is also possible to subscribe to a generic and sometimes free domain blacklist, that can be converted into a DNS RPZ blacklist. The

quality of such data is unfortunately often bad, which, in case of the DNS firewall, would result in a lot of needlessly blocked connections. As a result, this function was ultimately left out of our firewall. Even though we are still prepared to share our manually updated blacklist with others with similar interest.

V. Open Issues

During the implementation and testing of the DNS RPZ system, we have identified several issues that limit its practical usage. In this section, we present those issues and outline directions for future development that could help in solving them.

A. HTTPS Usage

The most significant issue with DNS RPZ is that it cannot redirect users when SSL/TLS is in use. The goal of the DNS firewall on web traffic is to inform users about their attempts to visit malicious or blocked domains, and to explain to them why those domains are dangerous. DNS RPZ works well when a simple HTTP is in use. It simply provides the user with an IP address of our server which serves the informative page for each request. However, this approach fails with HTTPS.

The users' web browser does not know about DNS redirection before contacting the target server. It assumes the IP address provided by the DNS system is valid and proceeds with standard SSL/TLS handshake. It specifies desired domain in Client hello message field SNI (Server Name Indication). The server then responds with Server hello message, where it needs to send the server certificate with a public key to establish the secure connection. However, it is impossible for the server to have a valid certificate for the blocked domain (by the design of certificate authority system), so the server includes a default certificate in the response. The browser then verifies the certificate and ends with name mismatch error, often displayed as *ssl_bad_cert_domain* or *ERR_CERT_COMMON_NAME_INVALID* together with a warning that someone is trying to impersonate the server. The user has to approve security exception to access the explanation page which is not the behavior we want to encourage in users. Another option for the server is to issue a self-signed certificate for each blocked domain, but the result of certificate verification would be virtually the same, only with a different error message.

The impact of this limitation will grow even more serious in the future with the concepts of HTTPS-by-default being introduced in major web browsers. A global solution would be a new type of certificates that would be issued for the sole purpose of domain blocking. Upon receiving such certificate, the browser would display a warning that the domain was blocked by the DNS service provider and would display a text message from the server with a detailed explanation why the domain was blocked. As this is a proposal of the future solution, the implementation details are open for discussion, but we assume that it is feasible to issue a certificate (or extended validation certificate) for company sub-domain in the form of *domainblock.company.com* which does not require any changes in current CA system. However, the browsers will have to implement a check for *domainblock* sub-domain to display the warning page. Another implementation problem would be showing the block reason so as not to be exploitable by external content loaded into a page with a valid certificate for any requested SNI. The browsers could restrict the message to plain text, or a new HTML meta tag could be introduced to list blocking reasons, and the server would only send the reason number to be translated into a message by the browser.

The cleanest solution would be to modify the DNS protocol itself. When creating the response, the DNS server can use response code *5 – Refused* [18] to indicate the request violates organization policies. However, the reason for domain blocking is still missing, and the response body would need to include text message explaining the block reason in a form that the receiving application could display to the user. The DNS protocol should specify the message format so that vendors can implement it. Furthermore, the DNS RPZ technology should adopt this response code as an option for domain blocking.

B. Domain Blacklist Sharing

We identified two main issues with the DNS firewall blacklist sharing. The most important issue is caused by the relatively small community and general reluctance to share the data. Currently, there are no organizations sharing their blacklists that we know of. The only remaining option to keep the blacklist up-to-date automatically being a paid subscription to a provider like FarsightSecurity [19], InfoBlox [15] or SpamHaus [20]. The blacklist sharing may be set up rather easily using the existing protocol for incremental DNS zone transfer (IXFR) and we are prepared to share our blacklist with other institutions implementing their DNS firewall.

Another significant issue comes from the trustworthiness of shared blacklist. A false positive domain on the blacklist, i.e., a legitimate domain which is on the blacklist by mistake, in an intrusion detection system can produce (in the worst case scenario) a flood of detections that will overwhelm the administrators. However, in the case of DNS RPZ, a false positive domain can lead to disruption of organization services as the domain is automatically blocked by the system. This problem is prominent especially in the field of phishing web pages which are often hosted on free hosting or exploit a misconfiguration of a legitimate page. Blocking on the level of the domain is then problematic as domains such as *forms.google.com* or the more general *google.com* can often be found in blacklist sharing platforms.

Manual processing of the shared domains is the obvious solution to avoid blocking of essential services, yet the human resources needed for its deployment are high. Our logging version of the DNS firewall represents a compromise with automatic logging of all accesses to suspicious domains. A human administrator can periodically process these accesses, and the top n of the domains can be manually checked and blocked. This approach introduces a delay in reaction to new threats but all communication is logged, and in the case of a

security incident, the victims can be traced from the firewall log.

C. Few Open Implementations

There is currently a lack of complex RPZ implementations that meet all the requirements for the DNS firewall, and that would enable its use by cybersecurity teams. The DNS firewall must consist of more parts besides the blocking itself – at least a management interface, a logging module and a landing page if users are to be redirected. These modules must also be reimplemented for every new firewall, which on one hand provides the freedom to fit them in the specific environment, but on the other hand consumes a significant amount of time and effort. We consider releasing our modules as an open-source to promote DNS firewall usage and development in the near future.

D. Easy Bypassing

Avoiding DNS-based blocking is relatively easy for the end user, he only needs to use a different DNS server (e.g., Google public DNS, OpenDNS) which can be changed with just a few clicks on Windows or by altering one line in Linux configuration file. However, by avoiding firewall for one blocked domain, the user also loses protection for every other malicious domain and exposes himself to threats.

Detection of the usage of external DNS servers are nowadays common part of anomaly detection systems [21] and the detection of users avoiding firewall is technically possible. It then depends on the authority of the company if they could order them to switch back to the DNS RPZ server.

VI. Conclusion

We consider the DNS firewall a suitable cybersecurity tool. It is effective, particularly when used along with other tools like IP blocking and automated malicious domains detection. We have implemented the firewall based on the only available open source technology, DNS Response Policy Zones. We present our experience with the implementation and deployment hoping that we will help to advance the idea of the DNS firewall as there is currently not much research being conducted in this area.

The DNS RPZ suffers from limitations that need to be taken into account. The most pressing issue is the inability to directly inform users of what caused the blocking of their communication attempt. While this does not preclude the main purpose of the DNS firewall – blocking access to malicious domains – obscurity does not serve well in this case. When a website is blocked without further clarification, users tend to try and circumvent the protection. It could also harm the firewall operator's trustworthiness in the long-term if websites get often blocked by mistake.

Of the four specified requirements, our DNS firewall fulfills completely the requirements for reliable domain blocking and event logging. With limitations, user informing is also possible, though only indirectly. Unfortunately, blacklist sharing is restricted only to commercially provided blacklists with no other suitable alternative. We continue researching possibilities for better DNS firewall implementation. Despite its shortcomings, we would prefer the DNS firewall to be based on DNS RPZ, as it has already been in development for eight years and provides a set of useful functions.

Acknowledgement

This research was supported by the Security Research Programme of the Czech Republic 2015 - 2020 (BV III / 1 VS) granted by the Ministry of the Interior of the Czech Republic under No. VI20172020070 Research of Tools for Cyber Situation Awareness and Decision Support of CSIRT Teams in the Protection of Critical Infrastructure. Martin Laštovička is Brno Ph.D. Talent Scholarship Holder – Funded by the Brno City Municipality. Special thanks belongs to Milan Čermák for his valuable insights on DNS communication.

References

[1] J. Nazario and T. Holz, "As the Net Churns: Fast-flux Botnet Observations," in *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*. IEEE, 2008, pp. 24–31.

[2] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich, "The Long "Taile" of Typosquatting Domain Names." in *USENIX Security Symposium*, 2014, pp. 191–206.

[3] G. Aceto and A. Pescapé, "Internet Censorship Detection: A Survey," *Computer Networks*, vol. 83, pp. 381–421, 2015.

[4] M. o. F. Czech Republic, "Dept 34 – State Oversight over Gambling," 2016.

[5] P. Vixie and V. Schryver, "DNS Response Policy Zones (RPZ)," Working Draft, IETF Secretariat, Internet-Draft draft-vixie-dnsop-dns-rpz-00, June 2018.

[6] P. Vixie, "Taking Back the DNS," http://www.circleid.com/posts/20100728_taking_back_the_dns/, 2010, accessed on 16.3.2018.

[7] H. M. Connery, "DNS: Response Policy Zone - A Case Study at DTU Environment," Tech. Rep., 2012. [Online]. Available: https://dnsrpz.info/spamhaus-rpz-case-study.pdf

[8] ——, "DNS Response Policy Zones Roadmap to Accellerate Adoption," Tech. Rep., 2013. [Online]. Available: https://dnsrpz.info/RPZ-Building-Momentum.pdf

[9] ——, "DNS Response Policy Zones History, Overview, Usage and Research."

[10] S. Crocker, D. Dagon, D. Kaminsky, D. D. McPherson, and P. Vixie, "Security and Other Technical Concerns Raised by the DNS Filtering Requirements in the PROTECT IP Bill," *White Paper*, vol. 6, p. 1, 2011.

[11] Afnic, "Consequences of DNS-based Internet Filtering," Tech. Rep.

[12] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," in *2010 Proceedings IEEE INFOCOM*, March 2010.

[13] M. Felegyhazi, C. Kreibich, and V. Paxson, "On the Potential of Proactive Domain Blacklisting," *LEET*, vol. 10, 2010.

[14] BlueCat, "BlueCat DNS," https://www.bluecatnetworks.com/products/dns/, 2018, accessed in March 2018.

[15] Infoblox, "DNS Firewall," https://www.infoblox.com/products/dns-firewall/, 2018, accessed in September 2018.

[16] P. Vixie, "DNS Firewalls In Action - RPZ vs. Spam," 2013.

[17] H. M. Connery, "RPZ Log Analysis," https://github.com/yesxorno/rpzla, 2013, accessed in March 2018.

[18] P. Mockapetris, "Domain Names - Implementation and Specification," RFC 1035, Internet Engineering Task Force, Nov. 1987.

[19] FarsightSecurity, "Newly Observed Domains," https://www.farsightsecurity.com/solutions/threat-intelligence-team/newly-observed-domains/, 2018, accessed in September 2018.

[20] SpamHaus, "Deteque RPZ Zones," https://www.deteque.com/app/uploads/2018/01/RPZ-Service-Overview.pdf, 2018, accessed in September 2018.

[21] F. Networks, "Flowmon ADS ISP 9.01.00 User Guide," 2017.